

Spring Configuration

Fedora 3.5 begins shifting the management and configuration of modules and components from the legacy Fedora Configuration syntax (.fcfg files) to the Spring 3 Framework. In this transitional release, Fedora will parse the modules and datasources in the fcfg as bean definitions, and then attempt to load all xml files in the spring configuration subdirectory as xml bean definitions. Bean definitions in this subdirectory will override fcfg configurations.

On this page:

- [Configuring the Server Module](#)
- [Configuring Modules](#)
- [Required Bean Configurations](#)
- [Newly Configurable Components](#)
 - [AttributeFinderModule \(3.5\)](#)
 - [JAX-RS \(3.6\)](#)
 - [JAX-WS \(3.6\)](#)
 - [PolicyEnforcementPoint \(3.6\)](#)
 - [Rebuilders \(3.5\)](#)
 - [TriplestoreConnector \(3.5\)](#)
 - [TripleGenerator \(3.5\)](#)
- [Spring Application Contexts in Fedora 3.5](#)
- [Spring Application Contexts in Fedora 3.6](#)

Folders	Name	Date modified	Type	Size
fedora-3.5	web	5/16/2011 11:07 AM	File Folder	
client	akubra-llstore.xml	5/24/2011 12:44 PM	XML Document	4 KB
data	policy-enforcement.xml	5/24/2011 12:44 PM	XML Document	4 KB
derby	README.txt	5/24/2011 12:44 PM	Text Document	9 KB
docs	triple-generator.xml	5/24/2011 12:44 PM	XML Document	3 KB
install				
pdp				
server				
access				
bin				
config				
spring				
web				

Configuring the Server Module

Beginning with Fedora 3.6, the Fedora Server module (ie, the implementation of `org.fcrepo.server.Server`) is configured in the Spring application context. This file is located under the Fedora web-app's WEB-INF directory; in the Tomcat unpacked by the installer, this will be in `$FEDORA_HOME/tomcat/webapps/fedora/WEB-INF/applicationContext.xml`. In this file, there is a Spring bean with an id of `"org.fcrepo.server.ServerConfiguration"` that reproduces the server module configurations that used to be available in the `fedora.fcfg` file. For example, specifying the hostname of your FCREpo server is done by changing the value of configuration Parameter of that bean whose first argument is `"fedoraServerHost"`. Other configurable parameters include:

- `datastreamExtensionMappingSource`
- `datastreamExtensionMappingId`
- `httpClientMaxConnectionsPerHost`
- `httpClientSocketTimeoutSecs`
- `fedoraServerHost`
- `datastreamExtensionMappingLabel`
- `fedoraServerPort`
- `fedoraRedirectPort`

... and others. The `applicationContext.xml` file is annotated with notes describing the configurable parameters and their acceptable values.

Configuring Modules

In the transition of 3.5, the interface of a module (also called the "role" in .fcfg files) is the name/id by which the module bean will be requested from the application context. Thus, if you are configuring the `DOManager` as a bean, you may write the following configuration to use the default implementation as a Spring bean:

```
<bean id="org.fcrepo.server.storage.DOManager"
name="org.fcrepo.server.storage.DOManager"
class="org.fcrepo.server.storage.DefaultDOManager"
destroy-method="shutdownModule" init-method="initModule" autowire-candidate="true" lazy-init="true">
<constructor-arg index="0">
<bean factory-bean="org.fcrepo.server.storage.DOManagerConfiguration" factory-method="getParameters" />
</constructor-arg>
<constructor-arg index="1" ref="org.fcrepo.server.Server" />
<constructor-arg index="2">
<bean factory-bean="org.fcrepo.server.storage.DOManagerConfiguration" factory-method="getRole" />
</constructor-arg>
</bean>
```

Take care to write and configure the shutdown/destroy methods for your module! Also note the configuration beans. It is a convention in 3.5 that a module configuration bean, if it is necessary, will be named (module name)Configuration. For example, the configuration in the example above, for **org.fcrepo.server.storage.DOManager**, is **org.fcrepo.server.storage.DOManagerConfiguration**. If you need such a configuration bean, you may use **org.fcrepo.server.config.ModuleConfiguration** bean to provide one like so:

```
<bean id="org.fcrepo.server.storage.DOManagerConfiguration"
  name="org.fcrepo.server.storage.DOManagerConfiguration"
  class="org.fcrepo.server.config.ModuleConfiguration"
  autowire-candidate="true">
<constructor-arg index="0">
<list>
<bean>
<constructor-arg type="java.lang.String" value="pidNamespace" />
<property name="value" value="changeme" />
</bean>
<bean>
<constructor-arg type="java.lang.String" value="storagePool" />
<property name="value" value="localDerbyPool" />
</bean>
<bean>
<constructor-arg type="java.lang.String" value="storageCharacterEncoding" />
<property name="value" value="UTF-8" />
</bean>
<bean>
<constructor-arg type="java.lang.String" value="defaultExportFormat" />
<property name="value" value="info:fedora/fedora-system:FOXML-1.1" />
</bean>
<bean>
<constructor-arg type="java.lang.String" value="gSearchRESTURL" />
<property name="value" value="http://localhost:8080/fedoragsearch/rest" />
</bean>
<bean>
<constructor-arg type="java.lang.String" value="gSearchUsername" />
<property name="value" value="exampleUsername" />
</bean>
<bean>
<constructor-arg type="java.lang.String" value="gSearchPassword" />
<property name="value" value="examplePassword" />
</bean>
</list>
</constructor-arg>
<constructor-arg index="1" type="java.lang.String" value="org.fcrepo.server.storage.DOManager" />
<constructor-arg index="2" type="java.lang.String" value="org.fcrepo.server.storage.DefaultDOManager" />
<constructor-arg index="3" type="java.lang.String" value="The interface to the storage subsystem." />
</bean>
```

Required Bean Configurations

As of the 3.5 release, there is only one uniformly required bean configuration: The *org.trippi.io.TripleIteratorFactory* bean must be included, with a destroy method mapped to "shutdown". If your repository has the Resource Index enabled, then you will also need an *org.fcrepo.server.resourceIndex.TripleGenerator* bean. Both of these beans are configured in \$FEDORA_HOME/server/config/spring/triple-generator.xml, which is present by default if you have run the installer.

Newly Configurable Components

AttributeFinderModule (3.5)

Beginning with version 3.5, the PolicyEnforcementPoint is configurable in Spring, allowing the configuration of new/additional AttributeFinderModules to support your XACML policy decisions.

JAX-RS (3.6)

Fedora 3.6 uses JAX-RS to manage its REST resources, and these resources are configured in Spring via Apache CXF. The configuration files for the JAX-RS resources are located in **\$FEDORA_HOME/server/config/spring/web/jaxrs/**. Resources configured here have access to beans in the applicationContext, but they do not have direct access to module beans loaded by the Server implementation (though they can request them directly from the Server).

JAX-WS (3.6)

Fedora 3.6 uses JAX-WS to manage its SOAP APIs. The configuration for these beans is in **\$FEDORA_HOME/server/config/spring/web/jaxws-impl.xml**. Like the JAX-RS beans, these APIs are configured in Spring via Apache CXF. The configured beans have access to beans in the applicationContext, but must request module beans from the Server bean.

PolicyEnforcementPoint (3.6)

In Fedora 3.6, more of the XACML policy enforcement machinery is exposed to configuration via Spring. While the PolicyEnforcementPoint implementations themselves are unlikely to be of general interest (unless you are writing an alternative authorization framework, like FESL), the exposure of the policy loading strategies as a configurable bean may be. The default implementation, which loads all policies, is **org.fcrepo.server.security.impl.SimplePolicyLoader**.

Rebuilders (3.5)

In version 3.5, the code that rebuilds Fedora's SQL database and Resource Index will look for all beans in the Spring context that implement **org.fcrepo.server.utilities.rebuild.Rebuilder**, and include them in its menu of rebuild options.

TriplestoreConnector (3.5)

Version 3.5 includes changes designed to allow more flexible configuration of TriplestoreConnectors. There is an example of this configuration in **\$FEDORA_HOME/server/config/spring/README.txt**.

TripleGenerator (3.5)

When objects are created (or purged) in Fedora with the Resource Index enabled, Fedora uses a TripleGenerator to determine which triples should be added or deleted from the index. In version 3.5, this component is configurable in Spring. Editing this configuration allows the addition of new content model-to-generator mappings for the default triple generator, **org.fcrepo.server.resourceIndex.ModelBasedTripleGenerator**.

Spring Application Contexts in Fedora 3.5

In Fedora 3.5, there are two "top level" application contexts: One manages the Spring components supporting the FESL authorization (AuthZ) filters, and the other is the Server context in which the Fedora modules (and other beans configured in **\$FEDORA_HOME/server/config/spring/**) are managed. These contexts are currently planned to be joined in a future version of Fedora.

Spring Application Contexts in Fedora 3.6

In Fedora 3.6, Spring-configured web components (like the FESL authorization filters) are loaded in the same application context as the Server bean. All the other modules are loaded, by default, in a context managed by the Server bean (ie, a child context).